



On the Soundness and Completeness of Equational Predicate Logics

George Tournakis

Technical Report CS-1998-08

November 2, 1998

Department of Computer Science
4700 Keele Street North York, Ontario M3J 1P3 Canada

On the soundness and completeness of Equational Predicate Logics

George Turlakis, York University
tgeorge@yorku.ca

Abstract

We present two different formalizations of Equational Predicate Logic, that is, first order logic that uses Leibniz’s substitution of “equals for equals” as a primary rule of inference.

We prove that both versions are sound and complete. A by-product of this study is an alternative proof to that contained in [GS3], that the “full” Leibniz rule is strictly stronger than the “no-capture” Leibniz rule, this result obtained here for a complete Logic.

We also show that under some reasonable conditions, propositional Leibniz, no-capture Leibniz, and a full-capture version are all equivalent, provided that the latter is restricted to act on universally valid premises whenever capture is allowed.

Introduction.

Equational Logic, as proposed by Dijkstra and Scholten [DSc], and propounded in the recent text and papers by Gries and Schneider [GS1, GS2, GS3], is finding its way in Computer Science curricula towards satisfying the “Discrete Math” requirements therein.

It is argued by its proponents that Equational Logic is the pedagogically proper setting to *do* proofs—i.e., work *within* the theory. It is also a fair observation that Equational Logic may be more cumbersome than “standard” approaches (e.g., [Sh, En, Man, Men]) when it becomes the object of study, in the metatheory—although this phenomenon depends on what semantics we wish to implement.[†]

The department of Computer Science at York University is currently using the equational approach to teaching logic to its first year undergraduates. Understandably, as [GS1] is addressed to undergraduates—and therefore it emphasizes the theory at the expense of the metatheory—it does not address the question of either *soundness* or *completeness* of its Equational Predicate Calculus.[‡] Indeed [GS1] leaves most of the syntax (and semantics) of the first order language where this Calculus is to be “spoken” up to the discretion of the instructor. The set of axioms in [GS1] has a high degree of redundancy, and the pivotal rule, Leibniz, is given under many guises which may be somewhat confusing. Moreover it is very hard to ascertain which rules of inference the authors of [GS1] wanted to be applicable to “universally valid” formulas only, and which ones to be applicable on any premises without restriction. (This uncertainty has triggered many spirited, but productive, e-mail exchanges with

[†]If “ A logically implies B ” means that every model of A is a model of B , then we have a more cumbersome metatheory (e.g., a cumbersome Deduction Theorem, [Sh, Men]). If on the other hand it means “every instantiation of the object variables that makes A true also makes B true”, then the metatheory is easier. In particular, a surprisingly easy to prove—and apply—Deduction Theorem becomes available ([Bou, En]).

[‡][GS2] cover these questions for the Propositional fragment of the Calculus.

colleagues at York University who, like the author, are teaching “Discrete Math” using the [GS1] text).

We aim in this paper to provide a precise syntax and semantics of a first order language and classical logic, as faithfully as possible to the spirit of the Equational Logic outlined in [GS1], and then to carefully develop the metatheory to the point that soundness can be derived in a self contained manner. We also address (and prove) completeness by showing that the Equational Logic of [GS1]—as it is systematically reincarnated here—is an extension of some standard first order logic (our “standard” yardstick is the approach taken in Shoenfield [Sh]).

As a by-product, we have reduced the very large number of Predicate Logic-specific axiom schemata offered in [GS1] (Chapters 8 and 9) to just two, and have reduced the additional “Leibniz” rule that is proposed in [GS1]—(8.12) in Chapter 8—to a derived rule status (*with a correction*). Naturally, these reductions will make Equational Calculus even more attractive, as they will reduce the burden placed on memory of both students and instructors—not to mention that it makes the metamathematics much easier.

Formalizing an informal logic is not easy. One has to try to understand what the “intentions” of the informal version were, and implement them *faithfully* through the syntactic apparatus. Unfortunately, it is in the nature of “informal intentions” to be vague.

In this instance we inferred, partly from the *practices in* [GS1] and partly from those in [GS3] the following two important points, which shaped our formalization (sections 1 and 2):

- (1) *Every rule is supposed to be applicable to arbitrary formulas*, not just to “universally valid” formulas. We draw this “inference” especially from [GS3] ([GS1] give short shrift to the concept of proofs *from a set of “nonlogical” axioms*).

Indeed, *all* the rules of inference in [GS3] are given in the format (for example) “(Equanimity) $\vdash P, \vdash P \equiv Q \longrightarrow \vdash Q$ ” where \longrightarrow is metalogical.

In [GS3], since *all* the rules are given in exactly this format—even the propositional rules—we had to conclude that the “unary” \vdash *must* mean “deducible from some (unspecified) nonlogical axioms”, otherwise this logic would *not* be usable as a tool for reasoning in any mathematical theory![†]

This usage of “ \vdash -notation” is widespread—e.g., [Sh] writes “ $\vdash A$ ” meaning that A is provable *from some (unspecified) nonlogical axioms*, but on occasion writes \vdash_T and $\vdash_{T'}$ if deductive power of two sets of nonlogical axioms, T, T' is being compared.

- (2) Should we, or should we not, restrict the rule “Leibniz” so that no “capture” of object variables occurs during its application?

[†]Obviously, one cannot, e.g., “prove $p \vee q$ from p ”, if every rule is *undefined* except when it applies to universally valid premises. With such rules, all theorems have to be universally valid—and $p \vee q$ is not.

The definitive answer was furnished by the proof-sketch of (9.16) (Meta-theorem “generalization”) in [GS1]. A pivotal step in the proof is that, once $A \equiv true$ is deduced, then, $(\forall x)A \equiv (\forall x)true$ follows by Leibniz.

Since A is expected to have x free in the “interesting case”, clearly, variable capture during an application of Leibniz is allowed in [GS1].[†]

The layout of the paper is as follows: Section 1 introduces the formal language. Section 2 builds a theory on top of this language (we refer to any such theory as an E-theory or E-logic). Section 3 introduces a host of metatheorems—that we will use towards the metatheoretical results of sections 4, 5 (on soundness and completeness, respectively) and 6—including the Deduction Theorem. The latter is heavily used in section 6, where the equivalence between our version of E-logic and the “original” version of [GS1] is proved. Most of the technical apparatus developed in this paper was aimed at making the proofs in section 6 “accessible”. The “main lemma” in sections 4–5 is lemma 5.3 that shows the eliminability of propositional variables. In section 7 we look at an alternative formalization, based on a weaker form of the Leibniz rule that avoids object variable “capture”. In the course of the discussion it emerges that “no-capture Leibniz” is equivalent to a “propositional” version of Leibniz and to a full capture version—which restricts its premises to be provable with no nonlogical axioms in those cases that capture occurs. It is also seen that either of these formulations is strictly weaker than “full” Leibniz (**Inf2** of section 2) providing a simple alternative proof to a result of [GS3].

A word on the proof of the Deduction Theorem is pertinent here. We opted to do it the “hard way”, that is, do it within the Equational Calculus setting rather than in an equivalent “standard” calculus.[‡] Handling the “Leibniz” rule in the present setting required a few lemmata and a second induction (on formulas) within the proof, making it uncharacteristically long (compare with the proof in [Sh]). This anecdote fortifies our belief that the Equational Calculus is best suited to do proofs, rather than study them.

1. Syntax

We think of a *first order language*, L , as triple $L = (V, \mathbf{Term}, \mathbf{Wff})$, where V is the *alphabet*, i.e., the set of basic syntactic objects (symbols) that we use to build “terms” and “formulas”. We start with a description of V , and then we describe the set of terms (**Term**) and the set of formulas (**Wff**).

Alphabet

[†]One cannot gloss over this by “dummy renaming”. Interestingly, in [GS3] it is stated that the Leibniz in [GS1] avoids capture. This must be a typo in [GS3], given the “evidence” to the contrary.

[‡]This would necessitate a metaproof of that equivalence, and in view of the presence of propositional variables *and* the rule of substitution in [GS1]—none of which are present in, say, [Sh, En, Men, Man, Ba]—would lengthen our exposition in a different, unnecessary, direction.

1. *Object variables.* An enumerable set x_1, x_2, \dots . We normally use the meta-symbols x, y, z, u, v, w with or without primes to stand for (object) variables.
2. *Boolean (or Sentential, or Propositional) variables.* An enumerable set v_1, v_2, \dots . We normally use the metasymbols p, q, r with or without primes to stand for Boolean variables.



These variables will be place-holders for *well-formed formulas*. This is a necessary contraction to allow the *Substitution* rule of inference, to simulate *axiom schemata*, since the latter are not allowed in the “propositional part” of [GS1].[†]



3. Equality (between “objects”—see below for its syntactic role), “=”.
4. Brackets, (and).
5. The Boolean or propositional connectives, $\equiv, \Rightarrow, \vee, \wedge, \neg$.
6. The quantifier, \forall (the existential quantifier, \exists , will be taken to be a meta-symbol, introduced definitionally).
7. The special Boolean (propositional) constants *true* and *false*.
8. A set of symbols (possibly empty) for *constants*. We normally use the meta-symbols a, b, c, d, e , with or without subscripts, to stand for constants unless we have in mind some alternative “standard” notation in selected areas of application of the 1st order logic (e.g., $\emptyset, 0, \omega$, etc.).
9. A set of symbols for *predicates* or *relations* (possibly empty) for each possible “arity” $n > 0$. We normally use P, Q, R with or without primes to stand for predicate symbols.
10. Finally, a set of symbols for *functions* (possibly empty) for each possible “arity” $n > 0$. We normally use f, g, h with or without primes to stand for function symbols.



1.1 Remark. *Any two* symbols mentioned in items 1–10 are distinct. Moreover (if they are built from simpler “sub-symbols”, e.g., x_1, x_2, x_3, \dots might really be $x|x, x||x, x|||x, \dots$), none is a *substring* (or *subexpression*) of any other.



Terms and Formulas

[†]In [GS3], while substitution is dropped in favour of adopting axiom schemata throughout, propositional variables are retained, presumably to facilitate statement and application of the Leibniz rule.

1.2 Definition. (*Terms*) The set of *terms*, **Term**, is the \subseteq -smallest set of strings or “expressions” over the alphabet 1–10 with the following two properties:

Any of the items in 1 or 8 (a, b, c, x, y, z , etc.) are in **Term**.

If f is a function[‡] of arity n and t_1, t_2, \dots, t_n are in **Term**, then so is the string $ft_1t_2\dots t_n$. \square

1.3 Definition. (*Atomic Formulas*) The set of *atomic formulas*, **Af**, contains precisely:

- 1) The symbols *true*, *false*, and every Boolean variable (that is, p, q, \dots).
- 2) The strings $t = s$ for every possible choice of terms t, s .
- 3) The strings $Pt_1t_2\dots t_n$ for every possible choices of n -ary predicates P (for all choices of $n > 0$) and all possible choices of terms t_1, t_2, \dots, t_n .

\square

1.4 Definition. (*Well-Formed Formulas*) The set of *well-formed formulas*, **Wff**, is the \subseteq -smallest set of strings or “expressions” over the alphabet 1–10 with the following properties:

- a) **Af** \subseteq **Wff**.
- b) If A, B are in **Wff**, then so are $(A \equiv B)$, $(A \Rightarrow B)$, $(A \wedge B)$, $(A \vee B)$.
- c) If A is in **Wff**, then so is $(\neg A)$.
- d) If A is in **Wff** and x is any object variable (*which may or may not occur (as a substring) in the formula A*), then the string $((\forall x)A)$ is also in **Wff**.

We say that A is the *scope* of $(\forall x)$.

\square



1.5 Remark. (1) A, B in the definition are so-called *syntactic* or meta-variables, used as *names* for (arbitrary) formulas. In general, we will let the letters A, B, C, D, E (with or without primes) be names for well-formed formulas, or just *formulas* as we often say.

(2) We introduce a meta-symbol (\exists) *solely in the metalanguage* via the definition “ $((\exists x)A)$ stands for, or abbreviates, $(\neg((\forall x)(\neg A)))$.”

(3) We often write more explicitly, $((\forall x)A[x])$ and $((\exists x)A[x])$ for $((\forall x)A)$ and $((\exists x)A)$. This is intended to draw attention to the variable x of A , which

[‡]We will omit the qualification “symbol” from terminology such as “function symbol”, “constant symbol”, “predicate symbol”.

has now become “bound”. Of course, notwithstanding the notation $A[x]$ (which only says that x *may* occur in A), x might actually not be a substring of A . In that case, intuitively, $((\forall x)A)$, $((\exists x)A)$ and A “mean” the same thing. This intuition is actually captured by the axioms and rules of inference, as we show in section 3 (3.7).

(4) In [GS1] the preferred metanotation for $((\forall x)A)$ and $((\exists x)A)$ is $(\forall x| : A)$ and $(\exists x| : A)$, respectively. They also have special (meta) notation for “bounded” quantification, that is $(\forall x|B : A)$ and $(\exists x|B : A)$ stand for $((\forall x)(B \Rightarrow A))$ and $((\exists x)(B \wedge A))$ respectively. In [GS1] the “ B ” part is called the “range” of the quantification.

(5) To minimize the use of brackets in the metanotation we adopt standard priorities, that is, \forall , \exists , and \neg have the highest, and then we have (in decreasing order of priority) \wedge , \vee , \Rightarrow , \equiv . All associativities are *right* (this is in variance with [GS1], but is just another acceptable—and common—convention of how to be sloppy in the metalanguage, and get away with it).

(6) The language just defined, **Wff**, is *one-sorted*, that is, it has a single *sort* or *type* of object variable. If one wants to admit several sorts, e.g., i_1, i_2, \dots for variables *intended* to take integer values *upon interpretation*, r_1, r_2, \dots intended to take real values, b_1, b_2, \dots intended to take Boolean values (true or false), etc., then we need a more elaborate definition of syntax.

We will need a different quantifier symbol for each sort, say, $\forall_i, \forall_r, \forall_b$, etc., and for each sort a dedicated “equality” predicate ($=_i, =_r, =_b$, etc.). Moreover, function symbols will have a sort or type associated with them (which will be the type of their *right field* in the intended interpretation). This will induce sorts (types) to all *terms*.

In the end of the day, we will turn around and use metanotation, suppressing the sort-subscripts (exactly as [GS1] do) and will let the context fend for the various (intended) sorts in the applications.

For this reason we do *not* burden the formal definition with many sorts. This deliberate omission will not restrict the applicability (to the multi-sorted case) of the few proof-theoretic and model-theoretic results included here.

It should be stated at once that should we feel the need to employ a *Boolean object variable type explicitly*, b_1, b_2, \dots , we would not allow the b_i to be the same as the v_1, v_2, \dots . The former are terms, the latter are (atomic) well formed formulas.



2. Axioms and Rules of Inference

The axioms and rules of inference will be chosen from [GS1]. We will not allow *schemata* in the so-called *propositional* (or *Boolean*) *axioms*, in order to agree with the present [GS1] edition. This decision will necessitate the presence of a *substitution rule* among the rules of inference. We will deviate somewhat in the choice of axioms outside the propositional domain, to maximize convenience. In particular, we will only adopt one among the proposed axioms in [GS1] (**Ax2**

below), and one easy to memorize theorem that occurs in [GS1] (namely, the “standard” axiom-schema of *instantiation* or *specialization*) as our axiom **Ax3**. This apparatus will be sufficient to ensure *completeness* as we will show.

2.1 Definition. (*Axioms and Axiom schemata*)

Ax1. Include all propositional calculus axioms from [GS1] (there is no harm to actually include all tautologies, as, e.g., Enderton [En] and Manin [Man] do).[†]

Ax2. (*Schema*) For every formulas A and B such that the variable x does *not* occur *free* in A , add

$$A \vee (\forall x)B \equiv (\forall x)(A \vee B)$$

NB. The above is (9.5) in [GS1]. It is not necessary to introduce the “range” part.

Ax3. (*Schema*) For every formula A , add

$$(\forall x)A \Rightarrow A[x := t], \text{ for any term } t$$

with a condition on t . The condition is that during substitution *no* variable of t (all such are free, of course) was “captured” by a quantifier. We say that “ t must be substitutable in x ”.

NB. We often see the above written as

$$(\forall x)A[x] \Rightarrow A[t]$$

or even

$$(\forall x)A \Rightarrow A[t]$$

where the presence of $A[x]$ (or $(\forall x)A$, or $(\exists x)A$) and $A[t]$ in the *same context* means that t replaces all x occurrences in A .

Ax4. (*Schema*) For *each* object variable x , add $x = x$.

Ax5. (*Leibniz’s characterization of equality—1st order version. Schema*) For any formula A , any object variable x and any term t that is *substitutable* in x (in A), add

$$x = t \Rightarrow A \equiv A[x := t]$$

NB. The above is written usually as

$$x = t \Rightarrow A[x] \equiv A[t]$$

or even

$$x = t \Rightarrow A \equiv A[t]$$

□

[†]Every propositional calculus formula is in **Wff**, of course, by definitions 1.2–1.4.



2.2 Remark. (1) In any formal setting that introduces many-sorts explicitly in the syntax, one will need as many versions of **Ax2–Ax5** as there are sorts.

An alternative formal approach is to have only *one* sort of object variable but introduce one-place predicates D_i, D_r, D_b , etc., in the alphabet—one for each sort—with the intended meanings: $D_i(x)$ “says” that x is an integer, $D_b(x)$ “says” that x is boolean, etc.

Thus, instead of, say, $(\forall_b x)A$ one would write $(\forall x)_{D_b}A$ —which is a short form of $(\forall x)(D_b(x) \Rightarrow A)$ —etc.

(2) Axioms **Ax4–Ax5** characterize equality between “objects”. [GS1], while they employ this type of “Leibniz” in their exposition, they do not seem to actually formally adopt the point of view of Predicate Calculus *with* equality. In any case, adding these two axioms makes this system (explicitly) applicable to mathematical theories such as number theory and set theory, and does not in any way burden what we would like to outline here. Indeed, we will only use these axioms in section 6.

(3) In **Ax3** and **Ax5** we imposed the condition that t must to be “substitutable” in x . I.e., it should not contain *free* variables that can be “captured” by quantification, as this would distort and invalidate the formula.

The following example is often given in the literature in order to illustrate what might happen after a mindless substitution:

Take A to stand for $(\exists y)\neg x = y$. Then $(\forall x)A[x] \Rightarrow A[y]$ is

$$(\forall x)(\exists y)\neg x = y \Rightarrow (\exists y)\neg y = y$$

and $x = y \Rightarrow A \equiv A[y]$ is

$$x = y \Rightarrow (\exists y)\neg x = y \equiv (\exists y)\neg y = y$$

neither of which, obviously, is universally valid.

The meta-remedy (advocated by many, for example, Shoenfield [Sh]) is to move the quantified variable(s) out of harm’s way, i.e., rename them so that no quantified variable in A has the same name as any (free, of course) variable in t .

This renaming is formally correct (i.e., it does not change the meaning of the formula) as we will see in the “variant” (meta)theorem in section 3. Of course, it is always possible to effect this renaming since we have countably many variables, and only finitely many appear free in t and A .

This trivial remedy allows us to render the conditions in **Ax3** and **Ax5** harmless. Essentially, a t is always “substitutable” *after renaming*.



2.3 Definition. (*Rules of Inference*) The following four are the *rules of inference*. These rules are relations on the set **Wff** and are written traditionally as “fractions”. We call the “numerator” the *premise(s)* and the “denominator” the *conclusion*.

We say that a rule of inference is *applied* to the formula(s) in the numerator, and that it *yields* (or *results in*) the formula in the denominator.

Any set $S \subseteq \mathbf{Wff}$ is *closed under* some rule of inference iff whenever the rule is applied to formulas in S , it also yields formulas in S .

Inf1. (*Substitution*) For any formulas A, B and any propositional variable p that may or may not occur in A (as a substring)

$$\frac{A}{A[p := B]}$$

with a condition. Namely, if p is in the scope of a quantifier $(\forall x)$, then x must *not* be free in B .

Inf2. (*Leibniz for formulas*) For any formulas A, B, C and any propositional variable p (which may or may not occur in C)

$$\frac{A \equiv B}{C[p := A] \equiv C[p := B]}$$

Inf3. (*Equanimity*) For any formulas A, B

$$\frac{A, A \equiv B}{B}$$

Inf4. (*Transitivity of \equiv*) For any formulas A, B, C

$$\frac{A \equiv B, B \equiv C}{A \equiv C}$$

□



2.4 Remark. (1) Let P be any 1-ary (“unary”) predicate and a any constant. Clearly, $(\forall x)(Px \Rightarrow Px) \Rightarrow Pa \Rightarrow Px$ is not universally valid, as an interpretation over \mathbb{N} with Px translated to say “ x is even” and a to be “the” 0 shows. Indeed a falsifying instance would be $(\forall x)(P^{\mathbb{N}}x \Rightarrow P^{\mathbb{N}}x) \Rightarrow P^{\mathbb{N}}0 \Rightarrow P^{\mathbb{N}}1$, where $P^{\mathbb{N}}$ denotes the “concrete” interpretation of P over \mathbb{N} that we have just suggested.

Yet, $(\forall x)(Px \Rightarrow p) \Rightarrow Pa \Rightarrow p$ is universally valid. This explains the “condition” for the substitution rule.

(2) *Leibniz for formulas* is the primary rule in [GS1] that supports “equational reasoning”. In standard approaches to logic it is not a primary rule, rather it appears as the well known “derived rule” (metatheorem) that if $\Gamma \vdash A \equiv B^\dagger$ and if we replace one or more occurrences of the subformula A of a formula D (here D is $C[p := A]$) by B , to obtain D' (that is $C[p := B]$), then $\Gamma \vdash D \equiv D'$. Shoenfield [Sh] calls this derived rule “the equivalence theorem”.

(3) [GS1] use “=” for “ \equiv ” in contexts where they want the symbol to act *conjunctionally*, rather than *associatively*, e.g., in successive steps of an equational-style proof.

[†]The meaning of the symbol \vdash is defined in 2.5.



We next define Γ -theorems, that is, formulas we can *prove from the set* of formulas Γ (this may be empty).

2.5 Definition. (Γ -theorems) The set of Γ -theorems, \mathbf{Thm}_Γ , is the \subseteq -smallest subset of \mathbf{Wff} that satisfies the following:

Th1. \mathbf{Thm}_Γ contains as a subset the *closure under $\mathbf{Inf1}^\dagger$* of: All the axioms in the group **Ax1** and all the *instances of axiom schemata Ax2–Ax5*.



We call the members of this closure *the logical axioms*.



Th2. $\Gamma \subseteq \mathbf{Thm}_\Gamma$. We call every member of Γ a *nonlogical axiom*.

Th3. \mathbf{Thm}_Γ is *closed under* each rule **Inf2–Inf4**.

The (meta)statement $A \in \mathbf{Thm}_\Gamma$ is traditionally written as $\Gamma \vdash A$, and we say that A is *proved from* Γ or that it is a Γ -theorem.

If $\Gamma = \emptyset$, then rather than $\emptyset \vdash A$ we write $\vdash A$. We often say in this case that A is *absolutely provable* (or provable with no nonlogical axioms). \square



2.6 Remark. The restriction in the application of **Inf1** means that we use it simply to eliminate propositional variables, in particular to turn the axioms of group **Ax1** to axiom-schemata. We also, in effect, eliminate its anachronistic presence from our logic, by “applying it up in front” in the course of a deduction. See also 3.24.



2.7 Definition. (Γ -proofs) A finite sequence A_1, \dots, A_n of members of \mathbf{Wff} is a Γ -proof iff every A_i , for $i = 1, \dots, n$ is one of

Pr1. A logical axiom (as in **Th1** above).

Pr2. A member of Γ .

Pr3. The result of a rule **Inf2–Inf4** applied to (an) appropriate formula(s) A_j with $j < i$.

\square

[†] “The closure of a set \mathcal{S} under an operation \mathcal{O} ” is the \subseteq -smallest set containing the members of \mathcal{S} and closed under \mathcal{O} .



2.8 Remark. (1) It is a well known result on inductive definitions that $\Gamma \vdash A$ is equivalent to “ A appears in some Γ -proof”—in the sense of the above definition—and also equivalent to “ A is at the end of some Γ -proof”.

(2) It follows from 2.7 that if each of A_1, \dots, A_n has a Γ -proof and B has an $\{A_1, \dots, A_n\}$ -proof, then B has a Γ -proof. Indeed, simply concatenate each of the given Γ -proofs (in any sequence). Append to the right of that sequence the given $\{A_1, \dots, A_n\}$ -proof (that ends with B). Clearly, the entire sequence is a Γ -proof, and ends with B .

We refer to this phenomenon as the *transitivity of \vdash* .

(3) If $\Gamma \subseteq \Delta$ and $\Gamma \vdash A$, then also $\Delta \vdash A$ as it follows from 2.5 or 2.7. In particular, $\vdash A$ implies $\Gamma \vdash A$ for any Γ .

(4) The inductive definition of theorems (2.5) allows one to prove properties of \mathbf{Thm}_Γ by *induction on theorems*. The equivalent *iterative definition*, via the concept of proof, allows us a different kind of induction, on the length of proofs.



3. Some Metatheorems (and the occasional theorem)

3.1 Metatheorem. For any formula A and any set of formulas Γ , $\Gamma \vdash A$ iff $\Gamma \vdash A \equiv true$.

Proof. (\rightarrow) In [GS1]

$$p \equiv p \equiv true \quad (1)$$

is an axiom (from group **Ax1**—we are using liberties granted to us by the symmetry and associativity of \equiv).

Thus, using **Inf1** and $[p := A]$ on (1), we get

$$\vdash A \equiv A \equiv true \quad (2)$$

that is, $A \equiv A \equiv true$ is a logical axiom. Now, 2.8(3) yields

$$\Gamma \vdash A \equiv A \equiv true \quad (3)$$

The assumption $\Gamma \vdash A$, (3) above, and *equanimity*, yield

$$\Gamma \vdash A \equiv true.$$

(\leftarrow) From [GS1], $\vdash true$, and hence, for any Γ , $\Gamma \vdash true$.

Thus, the assumption $\Gamma \vdash A \equiv true$, and *equanimity*, yield $\Gamma \vdash A$. \square

3.2 Theorem. $\vdash (\forall x)true \equiv true$.

Proof.

$$\begin{aligned}
 & (\forall x)true \\
 = & \langle \text{“Leibniz-rule” and } \vdash true \vee true \equiv true \rangle \\
 & (\forall x)(true \vee true) \\
 = & \langle \mathbf{Ax2} \rangle \\
 & true \vee (\forall x)true \\
 = & \langle \text{by } \vdash true \vee A \equiv true \rangle \\
 & true
 \end{aligned}$$

□

3.3 Metatheorem. For any Γ , any formula A and any choice of object variable x , $\Gamma \vdash A$ iff $\Gamma \vdash (\forall x)A$.

Proof.

(\rightarrow) We assume that $\Gamma \vdash A$. Thus,

$$\begin{aligned}
 & (\forall x)A \\
 = & \langle \text{“Leibniz-rule” and Metatheorem 3.1} \rangle \\
 & (\forall x)true \\
 = & \langle \mathbf{Theorem 3.2} \rangle \\
 & true
 \end{aligned}$$

(\leftarrow) We assume that $\Gamma \vdash (\forall x)A$. By **Ax3**, $\vdash (\forall x)Ax \Rightarrow A[x := x]$, that is, $\vdash (\forall x)A \Rightarrow A$, hence $\Gamma \vdash (\forall x)A \Rightarrow A$.



x is substitutable in x always.



By the hypothesis and modus ponens (a valid derived rule), we obtain $\Gamma \vdash A$. □

3.4 Corollary. (“Strong” Generalization) For any formula A , $A \vdash (\forall x)A$.

Proof. Take $\Gamma = \{A\}$ above. □



“Strong” implies that there is a weak generalization rule as well. In approaches to first order logic that contain only *propositional* rules of inference (e.g., [Bou], [En], where the only rule is modus ponens[†]), metatheorem 3.3 *requires* the constraint—in the only-if direction—that x is *not* free in any formula of Γ that is used in the deduction of A . In particular, $A \vdash (\forall x)A$ does *not* hold (see also 3.23(2)).

One then has the weak form “ $\vdash A$ iff $\vdash (\forall x)A$ ”.

Clearly then, the Leibniz rule does not hold in this approach to first order logic.



[†]We have here two non-propositional rules, **Inf1** and **Inf2**.

3.5 Corollary. (Substitution) $A[x_1, \dots, x_n] \vdash A[t_1, \dots, t_n]$, for any terms t_1, \dots, t_n .

Proof. Of course, the t_i must be “substitutable” in the respective variables. One can comfortably be silent about this in view of the variant theorem (3.10, below).

We illustrate the proof for $n = 2$. What makes it interesting is the requirement to have “simultaneous substitution”. To that end we first substitute into x_1 and x_2 *new* variables z, w —i.e., not occurring in either A or in the t_i . The proof is the following sequence.

$$\begin{array}{l}
 A[x_1, x_2] \\
 \langle \text{generalization} \rangle \\
 (\forall x_1)A[x_1, x_2] \\
 \langle \text{specialization; } x_1 := z \rangle \\
 A[z, x_2] \\
 \langle \text{generalization} \rangle \\
 (\forall x_2)A[z, x_2] \\
 \langle \text{specialization; } x_2 := w \rangle \\
 A[z, w] \\
 \langle \text{Now } z := t_1, w := t_2, \text{ in any order,} \\
 \text{is the same as “simultaneous substitution”} \rangle \\
 \langle \text{generalization} \rangle \\
 (\forall z)A[z, w] \\
 \langle \text{specialization; } z := t_1 \rangle \\
 A[t_1, w] \\
 \langle \text{generalization} \rangle \\
 (\forall w)A[t_1, w] \\
 \langle \text{specialization; } w := t_2 \rangle \\
 A[t_1, t_2]
 \end{array}$$

□

3.6 Theorem. For any formulas A, B such that the object variable x is not free in A , $\vdash A \Rightarrow (\forall x)B \equiv (\forall x)(A \Rightarrow B)$.

Proof. This is so by **Ax2**, $\vdash A \Rightarrow B \equiv \neg A \vee B$, and Leibniz for formulas (**Inf2**). □

3.7 Corollary. For any formulas A in which the object variable x is not free, $\vdash A \equiv (\forall x)A$.

Proof. We have $\vdash (\forall x)A \Rightarrow A$ anyway.

In the equational style we write

$$\begin{aligned}
 & A \Rightarrow (\forall x)A \\
 = & \langle \text{by 3.6} \rangle \\
 & (\forall x)(A \Rightarrow A) \\
 = & \langle \vdash A \Rightarrow A \equiv \text{true and Leibniz} \rangle \\
 & (\forall x)\text{true} \\
 = & \langle \text{by 3.2} \rangle \\
 & \text{true}
 \end{aligned}$$

□



The above Corollary fulfils the promise included in remark 1.5(3). The case for the metasyymbol \exists follows trivially by a sequence of equivalences:

$$\begin{aligned}
 & (\exists x)A \\
 & \langle \text{formalizing the "text" } (\exists x)A \rangle \\
 & \neg(\forall x)\neg A \\
 = & \langle \text{by 3.7 and Leibniz applied to } \neg p \rangle \\
 & \neg(\neg A) \\
 = & \\
 & A
 \end{aligned}$$

In particular, $\vdash (\forall x)\text{false} \equiv \text{false}$ and $\vdash (\exists x)\text{false} \equiv \text{false}$



3.8 Metatheorem. For any formulas A, B such that the object variable x is not free in A , $\Gamma \vdash A \Rightarrow B$ iff $\Gamma \vdash A \Rightarrow (\forall x)B$.

Proof. (\rightarrow) Assume $\Gamma \vdash A \Rightarrow B$.

Then $\Gamma \vdash (\forall x)(A \Rightarrow B)$ by 3.3, hence $\Gamma \vdash A \Rightarrow (\forall x)B$ by 3.6 (and equanimity).

(\leftarrow) Assume $\Gamma \vdash A \Rightarrow (\forall x)B$.

Then $\Gamma \vdash (\forall x)(A \Rightarrow B)$ by 3.6 (and equanimity), hence $\Gamma \vdash A \Rightarrow B$ by 3.3.

□

3.9 Corollary. For any formulas A, B such that the object variable x is not free in A , $\Gamma \vdash B \Rightarrow A$ iff $\Gamma \vdash (\exists x)B \Rightarrow A$.

3.10 Metatheorem. (The variant [or, dummy renaming] metatheorem) *For any formula $(\forall x)A$, if z does not occur in it (i.e., is neither free nor bound), then $\vdash (\forall x)A \equiv (\forall z)A[x := z]$.*

NB. We often write this (under the stated conditions) as $\vdash (\forall x)A[x] \equiv (\forall z)A[z]$.

Proof. We know that $\vdash (\forall x)A \Rightarrow A[x := z]$ (**Ax3**), since z is substitutable in x under the stated conditions. Thus, by 3.3,

$$\begin{aligned} & (\forall z)((\forall x)A \Rightarrow A[x := z]) \\ = & \langle \text{by 3.6, since } z \text{ is not free in } (\forall x)A \rangle \\ & (\forall x)A \Rightarrow (\forall z)A[x := z] \end{aligned}$$

Noting that x is not free in $(\forall z)A[x := z]$ and is substitutable in z (in $A[x := z]$), we can repeat the above argument to get \Leftarrow . \square

At this point we will be well advised to increase our metatheoretical *convenience*[†] by introducing two important results, namely, Post's tautology Theorem, and (Herbrand's) Deduction Theorem, in that order.

We first offer a careful definition of “tautologies” in Predicate Calculus.

3.11 Definition. (*Prime formulas in **Wff***) A formula $A \in \mathbf{Wff}$ is a *prime formula* iff it is any of

Pri1. Atomic

Pri2. A formula of the form $(\forall x)A$.

Let \mathcal{P} denote the set of *all* prime formulas in our language. Clearly, \mathcal{P} contains each propositional variable v_1, v_2, \dots . \square



That is, a prime formula has no “explicit” propositional connectives (in the case **Pri2** any connectives are hidden inside the scope of $(\forall x)$).

Clearly, $A \in \mathbf{Wff}$ iff A is a *Propositional Calculus formula over \mathcal{P}* (i.e., propositional variables will be all the strings in $\mathcal{P} - \{true, false\}$).



3.12 Definition. (*Tautologies in **Wff***) A formula $A \in \mathbf{Wff}$ is a *tautology* iff it is so when viewed as a Propositional Calculus formula over \mathcal{P} .

We call the set of all tautologies, as defined here, **Taut**. The symbol $\models_{\mathbf{Taut}} A$ says $A \in \mathbf{Taut}$. \square

The following generalizes 3.12 and will also be needed.

[†]The emphasis here is meant to draw a distinction between convenience and “power”. Whatever power we have got stems directly from the axioms and the rules of inference, and the definition of Γ -theorem or Γ -proof. All else that we add is just “convenience”.

3.13 Definition. (*Tautologically Implies, for formulas in \mathbf{Wff}*) Given formulas $B \in \mathbf{Wff}$ and $A_i \in \mathbf{Wff}$, for $i = 1, \dots, m$. We say that A_1, \dots, A_m *tautologically implies* B , in symbols $A_1, \dots, A_m \models_{\mathbf{Taut}} B$, iff (viewing the formulas as Propositional Calculus formulas over \mathcal{P}) *every truth assignment* to the prime formulas that makes each of the A_i true, also makes B true. \square



While a definition for an infinite set of premises is possible, we will not need it here.



3.14 Metatheorem. (Post's Tautology Theorem) *If $A \in \mathbf{Wff}$ is a tautology in the sense of 3.12, then $\vdash A$ with a proof that uses only (substitution instances—under **Inf1**—of) axioms in the list **Ax1**.*

Proof. First, we note the following equivalences. (In the course of this proof, p, q, r stand for *any member* of $\mathcal{P} - \{true, false\}$, not just for propositional variables of our language.)

$$\begin{aligned} \models_{\mathbf{Taut}} true &\equiv \neg p \vee p, \text{ and also } \vdash true \equiv \neg p \vee p \\ \models_{\mathbf{Taut}} false &\equiv \neg p \wedge p, \text{ and also } \vdash false \equiv \neg p \wedge p \\ \models_{\mathbf{Taut}} p \Rightarrow q &\equiv \neg p \vee q, \text{ and also } \vdash p \Rightarrow q \equiv \neg p \vee q \\ \models_{\mathbf{Taut}} p \wedge q &\equiv \neg(\neg p \vee \neg q), \text{ and also } \vdash p \wedge q \equiv \neg(\neg p \vee \neg q) \\ \models_{\mathbf{Taut}} (p \equiv q) &\equiv ((p \Rightarrow q) \wedge (q \Rightarrow p)), \text{ and also } \vdash (p \equiv q) \equiv ((p \Rightarrow q) \wedge (q \Rightarrow p)) \end{aligned}$$

where \vdash , everywhere above, is achieved by staying within the axiom group **Ax1**.

Thus, if we *transform* A into A' by applying any sequence of the above equivalences to *eliminate* all occurrences of *true* and *false* and all the connectives except \neg and \vee , then we have, on the one hand, that $\models_{\mathbf{Taut}} A'$ and on the other hand (by the Leibniz rule) that $\vdash A \equiv A'$.

Thus, by equanimity, it suffices to prove $\vdash A'$.



A better way to say all this is that, *without loss of generality*, we assume that the only connectives in A are among \vee and \neg and that the constants *true* and *false* do not occur.

Moreover, since $\vdash A \vee A \equiv A$, we may assume *without loss of generality* that A is a string $A_1 \vee \dots \vee A_n$ with $n \geq 2$, so that none of the A_i is a formula $C \vee D$. We are assuming metanotational abbreviations when it comes to bracketing.

Let us call an A_i *reducible* iff it has the form $\neg(C \vee D)$ or $\neg(\neg C)$. Otherwise it is *irreducible*. Thus, the only possible irreducible A_i have the form p or $\neg p$ (where $p \in \mathcal{P} - \{true, false\}$). We say that A is *irreducible* iff all the A_i are.

We define the *reducibility degree* of A_i to be the number of \neg or \vee connectives in it, *not counting a possible leftmost \neg* . The *reducibility degree* of A is the sum of the *reducibility degrees* of all its A_i .

So let $\models_{\mathbf{Taut}} A$, where A is the string $A_1 \vee \dots \vee A_n$, $n \geq 2$, where none of the A_i is a formula $C \vee D$, and prove (following Shoenfield) by induction on the reducibility degree of A that $\vdash A$.



For the basis, let A be an irreducible tautology. It must be that A is a string of the form “ $\dots \vee p \vee \dots \neg p \vee \dots$ ” for some p , otherwise (if no p appears both “positively” and “negatively”) we can find a truth-assignment that makes A false—a contradiction (assign false to p ’s that occur positively only, and true to those that occur negatively only).

Since we have both semantic and syntactic commutativity, let us simplify notation and assume that A is $p \vee \neg p \vee B$. We know enough from [GS1] to conclude that $\vdash p \vee \neg p \vee B$.

Let then A_i be reducible. Again, since we have both semantic and syntactic commutativity, let us assume without restricting generality that $i = 1$.

We have two cases:

(1) A has the form $\neg(\neg C) \vee D$ (i.e., A_1 is the string $\neg(\neg C)$). Clearly $C \vee D$ is also a tautology, of a *lower reducibility degree*, hence provable (by Induction Hypothesis—in short, I.H.). Thus,

$$\begin{aligned} & \text{true} \\ &= \langle \text{by I.H.} \rangle \\ & \quad C \vee D \\ &= \langle \text{Leibniz} \rangle \\ & \quad \neg(\neg C) \vee D \end{aligned}$$

(2) A has the form $\neg(C \vee D) \vee E$, that is $(C \vee D) \Rightarrow E$ (i.e., A_1 is the string $\neg(C \vee D)$). Clearly $C \Rightarrow E$ and $D \Rightarrow E$ are both tautologies of lower (each) reducibility degree than the original.

By I.H., we have $\vdash C \Rightarrow E$ and $\vdash D \Rightarrow E$. By material in [GS1] (“definition by cases”), we have $\vdash (C \vee D) \Rightarrow E$. \square

3.15 Corollary. *If $A_1, \dots, A_n \models_{\mathbf{Taut}} B$, then $A_1, \dots, A_n \vdash B$.*

Proof. It is an easy semantic exercise to see that

$$\models_{\mathbf{Taut}} A_1 \Rightarrow \dots \Rightarrow A_n \Rightarrow B.$$

By 3.14,

$$\vdash A_1 \Rightarrow \dots \Rightarrow A_n \Rightarrow B$$

hence

$$A_1, \dots, A_n \vdash A_1 \Rightarrow \dots \Rightarrow A_n \Rightarrow B \tag{1}$$

Applying modus ponens n times to (1) we get

$$A_1, \dots, A_n \vdash B$$

\square



The above corollary is very convenient. We normally apply it as follows (compare with remark 2.8(2)): If each of A_1, \dots, A_n has a Γ -proof (equivalently, is a Γ -theorem) then so does (equivalently, is) B , if $A_1, \dots, A_n \models_{\mathbf{Taut}} B$.



3.16 Corollary. *The set of Γ -theorems, \mathbf{Thm}_Γ , is the \subseteq -smallest subset of \mathbf{Wff} that satisfies the following:*

- (i) \mathbf{Thm}_Γ contains as elements all logical axioms
- (ii) $\Gamma \subseteq \mathbf{Thm}_\Gamma$
- (iii) \mathbf{Thm}_Γ is closed under any rule

$$\frac{A_1, \dots, A_n}{B}$$

such that $A_1, \dots, A_n \models_{\mathbf{Taut}} B$

- (iv) \mathbf{Thm}_Γ is closed under rule **Inf2** (Leibniz).

Proof. **NB.** This statement is the same as that in definition 2.5, with a difference only in the third clause (compare (iii) here with **Th3** in 2.5).

For a proof, let us call $\widehat{\mathbf{Thm}}_\Gamma$ the set inductively defined here, and show that $\widehat{\mathbf{Thm}}_\Gamma = \mathbf{Thm}_\Gamma$.

(\subseteq) We do induction on $\widehat{\mathbf{Thm}}_\Gamma$. We note that \mathbf{Thm}_Γ satisfies

- (1) it contains the same initial objects as $\widehat{\mathbf{Thm}}_\Gamma$ (compare **Th1–Th2** with (i)–(ii)),
- (2) it is closed under **Inf2**,
- (3) it is closed under rules (iii), by 3.15 and *transitivity of \vdash* .

The set $\widehat{\mathbf{Thm}}_\Gamma$ being the \subseteq -smallest with properties (1)–(3), we are done.

(\supseteq) We do induction on \mathbf{Thm}_Γ . We note that $\widehat{\mathbf{Thm}}_\Gamma$ satisfies

- (4) it contains the same initial objects as \mathbf{Thm}_Γ (compare (i)–(ii) with **Th1–Th2**),
- (5) it is closed under **Inf2**,
- (6) it is closed under rules **Inf3–Inf4** since each of them has its conclusion tautologically implied by the premises(s), so, it is a rule of type (iii).

The set \mathbf{Thm}_Γ being the \subseteq -smallest with properties (4)–(6), we are done once more. \square



Neither of **Inf1**, **Inf2** has the form of a tautological implication.



3.17 Corollary. $\Gamma \vdash A$ iff there is a sequence of formulas A_1, \dots, A_n such that A_n is B (identical strings), and each A_i satisfies one of the following conditions:

- (a) A_i is a logical axiom,
- (b) $A_i \in \Gamma$,
- (c) For some A_{j_m} with $j_m < i$ ($m = 1, \dots, r$), $A_{j_1}, \dots, A_{j_r} \models_{\mathbf{Taut}} A_i$,
- (d) A_i is the result of **Inf2** on some A_j ($< i$).

3.18 Lemma. For any formulas A and B , $\vdash (\forall x)(A \Rightarrow B) \Rightarrow (\forall x)A \Rightarrow (\forall x)B$.

Proof. We write an annotated proof (omitting redundant brackets):

- | | | |
|-----|--|----------------------------|
| (1) | $(\forall x)(A \Rightarrow B) \Rightarrow A \Rightarrow B$ | ⟨ Ax3 ⟩ |
| (2) | $(\forall x)A \Rightarrow A$ | ⟨ Ax3 ⟩ |
| (3) | $(\forall x)(A \Rightarrow B) \Rightarrow (\forall x)A \Rightarrow A$ | ⟨(2) and 3.15⟩ |
| (4) | $(\forall x)(A \Rightarrow B) \Rightarrow (\forall x)A \Rightarrow B$ | ⟨(1), (3), and 3.15⟩ |
| (5) | $(\forall x)((\forall x)(A \Rightarrow B) \Rightarrow (\forall x)A \Rightarrow B)$ | ⟨(4), and 3.4⟩ |
| (6) | $(\forall x)(A \Rightarrow B) \Rightarrow (\forall x)A \Rightarrow (\forall x)B$ | ⟨(5), 3.6, and equanimity⟩ |

□

3.19 Lemma. For any A, B, C such that x is not free in A ,

$$A \Rightarrow (B \equiv C) \vdash A \Rightarrow ((\forall x)B \equiv (\forall x)C).$$

Proof. We write an annotated proof:

- | | | |
|-----|--|---|
| (1) | $A \Rightarrow (B \equiv C)$ | ⟨hypothesis⟩ |
| (2) | $A \Rightarrow (B \Rightarrow C)$ | ⟨(1) and 3.15⟩ |
| (3) | $A \Rightarrow (\forall x)(B \Rightarrow C)$ | ⟨ x not free in A . (2) and 3.8 used⟩ |
| (4) | $(\forall x)(B \Rightarrow C) \Rightarrow (\forall x)B \Rightarrow (\forall x)C$ | ⟨3.18⟩ |
| (5) | $A \Rightarrow (\forall x)B \Rightarrow (\forall x)C$ | ⟨(3), (4), and 3.15⟩ |

A similar proof establishes that $A \Rightarrow (B \equiv C) \vdash A \Rightarrow (\forall x)C \Rightarrow (\forall x)B$.

Since

$$A \Rightarrow (\forall x)B \Rightarrow (\forall x)C, A \Rightarrow (\forall x)C \Rightarrow (\forall x)B \models_{\mathbf{Taut}} A \Rightarrow ((\forall x)B \equiv (\forall x)C)$$

we are done by 3.15. □

3.20 Metatheorem. (The Deduction Theorem) If A is closed and $\Gamma, A \vdash B$, then $\Gamma \vdash A \Rightarrow B$.



NB. Γ, A means $\Gamma \cup \{A\}$. A converse of the metatheorem is also true, without any restriction on A : That is, $\Gamma \vdash A \Rightarrow B$ implies $\Gamma, A \vdash B$. This follows by modus ponens. We only use the Deduction theorem in section 6.



Proof. The proof is by induction on Γ, A -theorems, as these are characterized in corollary 3.16.

Basis. Let B be logical. Then, $\vdash B$, and hence $\Gamma \vdash B$.

Trivially, $B \models_{\mathbf{Taut}} A \Rightarrow B$, hence, by 3.15 and the transitivity of \vdash , $\Gamma \vdash A \Rightarrow B$.

The same reasoning applies if $B \in \Gamma$, since then $\Gamma \vdash B$ as well.

Finally, if B is the same string as A , then $\vdash A \Rightarrow B$, hence $\Gamma \vdash A \Rightarrow B$.

There are only two induction steps, by 3.16.

1. (*Leibniz*) Let $\Gamma, A \vdash C \equiv D$, and B be the string $E[p := C] \equiv E[p := D]$ for some formula E and propositional variable p .

By I.H., $\Gamma \vdash A \Rightarrow (C \equiv D)$.

By induction on the formula E , we show that

$$\Gamma \vdash A \Rightarrow (E[p := C] \equiv E[p := D]) \quad (1)$$

Basis. If E is any of *true*, *false*, q , $Pt_1 \dots t_n$, or $t_1 = t_2$, then the right hand side of \Rightarrow in (1) reads $E \equiv E$ and the result is trivial (from $\vdash E \equiv E$, and 3.15).

If E is p , then the right hand side of \Rightarrow in (1) reads $C \equiv D$, and the sought result is identical to the I.H.

Let then E be $\neg G$ (in least parenthesized metanotation). By I.H.,

$$\Gamma \vdash A \Rightarrow (G[p := C] \equiv G[p := D])$$

thus (1) follows from 3.15 and

$$A \Rightarrow (G[p := C] \equiv G[p := D]) \models_{\mathbf{Taut}} A \Rightarrow (\neg G[p := C] \equiv \neg G[p := D]).$$

Let next E be $G \circ H$ (in least parenthesized metanotation, where \circ is any of $\equiv, \vee, \wedge, \Rightarrow$). By I.H.,

$$\Gamma \vdash A \Rightarrow (G[p := C] \equiv G[p := D])$$

and

$$\Gamma \vdash A \Rightarrow (H[p := C] \equiv H[p := D])$$

thus (1) follows from 3.15 and

$$\begin{aligned} A \Rightarrow (G[p := C] \equiv G[p := D]), A \Rightarrow (H[p := C] \equiv H[p := D]) \\ \models_{\mathbf{Taut}} A \Rightarrow ((G \circ H)[p := C] \equiv (G \circ H)[p := D]) \end{aligned}$$

Finally, let E be $(\forall x)G$. By I.H., $\Gamma \vdash A \Rightarrow (G[p := C] \equiv G[p := D])$.

Since A is closed, x is not free in A . Thus, by 3.19,

$$\Gamma \vdash A \Rightarrow (((\forall x)G)[p := C] \equiv ((\forall x)G)[p := D])$$

that is, (1) holds.

2. (*Tautological implication rules*) Let $\Gamma, A \vdash A_i$ (for $i = 1, \dots, r$) and

$$A_1, \dots, A_r \models_{\mathbf{Taut}} B. \quad (2)$$

By I.H., $\Gamma \vdash A \Rightarrow A_i$ (for $i = 1, \dots, r$). On the other hand, (2) yields[†]

$$A \Rightarrow A_1, \dots, A \Rightarrow A_r \models_{\mathbf{Taut}} A \Rightarrow B$$

Thus (3.15 and transitivity of \vdash), $\Gamma \vdash A \Rightarrow B$.

□

For easy applicability of 3.20 we need some additional results.

3.21 Metatheorem. (Metatheorem on constants) *Let e be a new constant symbol, added to the alphabet V of the language L resulting to the language L' . Let Γ' denote the theory over L' that has precisely the same nonlogical axioms as Γ .*[‡]

Then, $\Gamma' \vdash A[e]$ implies $\Gamma \vdash A[x]$, for any variable x that appears nowhere in the Γ' -proof, as either a free or a bound variable.

Proof. We do induction on Γ' -theorems, using 3.16.

Basis. If $A[e]$ is logical, then so is $A[x]$. $A[e]$ cannot be nonlogical in Γ' .

Tautological implication. If $\Gamma' \vdash B_i[e]$, for $i = 1, \dots, n$, and $B_1[e], \dots, B_n[e] \models_{\mathbf{Taut}} A[e]$, then also $B_1[x], \dots, B_n[x] \models_{\mathbf{Taut}} A[x]$. By the I.H. (i.e., $\Gamma \vdash B_i[x]$, for $i = 1, \dots, n$), we are done in this case.

Leibniz. Let $\Gamma' \vdash B[e] \equiv C[e]$ and $A[e]$ be $D[e][p := B[e]] \equiv D[e][p := C[e]]$. Clearly, $A[x]$ is $D[x][p := B[x]] \equiv D[x][p := C[x]]$, and we are done by I.H., for one last time. □

3.22 Corollary. *Let e be a new constant symbol, added to the alphabet V of the language L resulting to the language L' . Let Γ' denote the theory over L' that has precisely the same nonlogical axioms as Γ .*

Then, for any variable y , $\Gamma' \vdash A[e]$ iff $\Gamma \vdash A[y]$.

Proof. (if) Trivially, $\Gamma' \vdash A[y]$, hence $\Gamma' \vdash A[e]$ by 3.5.

(only-if) Choose a variable x at first with the same restrictions as in 3.21, to obtain $\Gamma \vdash A[x]$.

Now apply 3.5 (and 3.10, to avoid substitutability concerns) to get $\Gamma \vdash A[y]$. □



3.23 Remark. (1) 3.22 adds flexibility to 3.20.

$$\Gamma \vdash (A \Rightarrow B)[x_1, \dots, x_n] \quad (1)$$

[†]Let a truth assignment make the left of $\models_{\mathbf{Taut}}$ true. The “hard case” is when it makes A true at the same time. This forces all the A_i to be true, and by (2), B to be true.

[‡]Thus, in particular, Γ' is “aware” of some additional facts. For example, $\Gamma' \vdash e = e$, whereas “ $e = e$ ” is meaningless in (the language of) Γ .

where $[x_1, \dots, x_n]$ is the list of *all free variables in A* , is equivalent (by 3.22) to

$$\Gamma' \vdash (A \Rightarrow B)[e_1, \dots, e_n] \quad (2)$$

where e_1, \dots, e_n are “new” constants.

Since $A[e_1, \dots, e_n]$ is closed, by 3.20, proving

$$\Gamma', A[e_1, \dots, e_n] \vdash B[e_1, \dots, e_n]$$

establishes (2), hence (1).

In practice, one does not perform this step explicitly, but ensures that throughout the Γ, A -proof, whatever free variables were present in A “behave like constants”.

(2) In some expositions the Deduction Theorem is *not* constrained by requiring that A be closed (e.g., [Bou], and more recently [En]).

Which version is right? They both are. If all the rules of inference are “propositional” (e.g., as in [Bou, En], who only employ modus ponens), then the Deduction theorem is unconstrained. If, on the other hand, the rules of inference manipulate object variables via quantification, then one cannot avoid constraining the application of the deduction Theorem, lest one wants to derive (the invalid) $\vdash A \Rightarrow (\forall x)A$ from the valid $A \vdash (\forall x)A$.

How do [Bou, En] cope with this issue? Well, in their approach they do *not* have $A \vdash (\forall x)A$. They have instead the weaker $\Gamma \vdash A$ iff $\Gamma \vdash (\forall x)A$, *with the condition* (in the only-if direction) that x not be free in any formula of Γ used in the deduction of A .

(3) This divergence of approach in choosing rules of inference has some additional repercussions.

One has to be careful in defining the semantic counterpart of \vdash , namely, \models (see next section). One wants the two symbols to “track each other” faithfully (Gödel’s completeness theorem).[†]



We state a metatheorem that allows some additional flexibility in substitutions. It shows that we *can* apply substitution (to a known absolute theorem) *post facto*.

3.24 Metatheorem. (Substitution in absolute theorems) *If $\vdash A$, then $\vdash A[p := W]$ for any formula W and variable p .*

Proof. Induction on theorems.

Basis. A is a (logical) axiom. Then $A[p := W]$ is as well (by 2.5), thus $\vdash A[p := W]$.

There are only two induction steps, by 3.16.

[†]In [Men] \models is defined inconsistently with \vdash .

1. (*Leibniz*) A is $C[s := D] \equiv C[s := E]$ and $\vdash D \equiv E$ (we may view s as a new variable, in particular, different from p).
By I.H., $\vdash D[p := W] \equiv E[p := W]$, hence $\vdash R[s := D[p := W]] \equiv R[s := E[p := W]]$ by Leibniz, where R is $C[p := W]$.
2. $A_1, \dots, A_m \models_{\mathbf{Taut}} A$ and $\vdash A_i$ ($i = 1, \dots, m$). By I.H., $\vdash A_i[p := W]$ ($i = 1, \dots, m$), but also $A_1[p := W], \dots, A_m[p := W] \models_{\mathbf{Taut}} A[p := W]$.

□

4. Soundness

The easiest way to introduce Tarski semantics is to follow Shoenfield.

4.1 Definition. Given a language $L = (V, \mathbf{Term}, \mathbf{Wff})$, a *structure* $\mathfrak{M} = (M, \mathcal{I})$ *appropriate for* L is such that $M \neq \emptyset$ is a set (the “domain”) and \mathcal{I} is a mapping that assigns

- (1) to each constant a of V a unique member $a^{\mathcal{I}} \in M$
- (2) to each function f of V —of arity n —a unique (total) function $f^{\mathcal{I}} : M^n \rightarrow M$
- (3) to each predicate P of V —of arity n —a unique set $P^{\mathcal{I}} \subseteq M^n$
- (4) to each propositional variable p of V a unique member $p^{\mathcal{I}}$ of the two element set $\{\mathbf{t}, \mathbf{f}\}$ (we understand \mathbf{t} as “true” and \mathbf{f} as “false”)
- (5) moreover we set $true^{\mathcal{I}} = \mathbf{t}$ and $false^{\mathcal{I}} = \mathbf{f}$, where the use of “=” here is metamathematical (equality on $\{\mathbf{t}, \mathbf{f}\}$).

□

We would now like to take a formula A , transform each one of its syntactic ingredients—except free variables— \mathcal{S} into its “concrete” counterpart $\mathcal{S}^{\mathcal{I}}$ to finally obtain $A^{\mathcal{I}}$. We could then say that A is *valid in* \mathfrak{M} , and write $\models_{\mathfrak{M}} A$ to mean that $A^{\mathcal{I}}$ is true for *all* values (from M) “plugged into” its free variables. This can be done as follows (among other ways).

4.2 Definition. Given L and a structure $\mathfrak{M} = (M, \mathcal{I})$ appropriate for L . $L(\mathfrak{M})$ denotes the language obtained from L by *adding* in V a unique *name* \hat{i} for each object $i \in M$. This amends both sets \mathbf{Term} , \mathbf{Wff} into $\mathbf{Term}(\mathfrak{M})$, $\mathbf{Wff}(\mathfrak{M})$. Members of the latter sets are called \mathfrak{M} -terms and \mathfrak{M} -formulas respectively.

We extend \mathcal{I} to the new constants: $\hat{i}^{\mathcal{I}} = i$ for all $i \in M$ (where the metamathematical “=” is that on M). □



All we have done here is to allow ourselves to do substitutions like $[x := i]$ formally. We do instead, $[x := \hat{i}]$.



One next gives “meaning” to all *closed* terms in $L(\mathfrak{M})$.

4.3 Definition. For *closed* terms t in $\mathbf{Term}(\mathfrak{M})$ we define the symbol $t^{\mathcal{I}} \in M$ inductively:

- (1) If t is any of a (original constant) or \hat{i} (imported constant), then $t^{\mathcal{I}}$ has already been defined.
- (2) If t is the string $ft_1 \dots t_n$, where f is n -ary, and t_1, \dots, t_n are closed \mathfrak{M} -terms, we define $t^{\mathcal{I}}$ to be the object (of M) $f^{\mathcal{I}}(t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}})$.

□

Finally, we give meaning to all *closed \mathfrak{M} -formulas* (that is, \mathfrak{M} -sentences).

4.4 Definition. For *closed* formulas A in $\mathbf{Wff}(\mathfrak{M})$ we define the symbol $A^{\mathcal{I}}$ inductively. In all cases, $A^{\mathcal{I}} \in \{\mathbf{t}, \mathbf{f}\}$.

- (1) If A is any of p or *true* or *false*, then $A^{\mathcal{I}}$ has already been defined.
- (2) If A is the string $t = s$, where t and s are closed \mathfrak{M} -terms, then $A^{\mathcal{I}} = \mathbf{t}$ iff $t^{\mathcal{I}} = s^{\mathcal{I}}$ (again, the last two occurrences of $=$ refer to equality on $\{\mathbf{t}, \mathbf{f}\}$ and M respectively).
- (3) If A is the string $Pt_1 \dots t_n$, where P is an n -ary predicate and the t_i are closed \mathfrak{M} -terms, then $A^{\mathcal{I}} = \mathbf{t}$ iff $(t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) \in P^{\mathcal{I}}$.
- (4) If A is any of $\neg B, B \wedge C, B \vee C, B \Rightarrow C, B \equiv C$, then $A^{\mathcal{I}}$ is determined by the usual truth tables using the values $B^{\mathcal{I}}$ and $C^{\mathcal{I}}$.
- (5) If A is $(\forall x)B$, then $A^{\mathcal{I}} = \mathbf{t}$ iff $(B[x := \hat{i}])^{\mathcal{I}} = \mathbf{t}$ for all $i \in M$.

□

4.5 Definition. Let $A \in \mathbf{Wff}$ and \mathfrak{M} be a structure as above. An \mathfrak{M} -instance of A is a sentence $\bar{A} \in \mathbf{Wff}(\mathfrak{M})$ obtained from A by substituting various constants \hat{i}, \hat{j}, \dots into *all* the free variables of A .

We say that A is *valid in \mathfrak{M}* , or that \mathfrak{M} is a *model of A* —in symbols $\models_{\mathfrak{M}} A$ —iff for *all possible \mathfrak{M} -instances \bar{A} of A* we have $\bar{A}^{\mathcal{I}} = \mathbf{t}$.

For any set of formulas Γ from \mathbf{Wff} , $\models_{\mathfrak{M}} \Gamma$ denotes the sentence “ \mathfrak{M} is a model of Γ ”, and means that *for all $A \in \Gamma$, $\models_{\mathfrak{M}} A$* .

A formula A is *universally valid* (we often say just *valid*) iff *every structure appropriate for the language is a model of A* . We then simply write $\models A$. □

4.6 Definition. We say that Γ *logically implies* A , in symbols $\Gamma \models A$, to mean that *every model of Γ is also a model of A* . □




This is the correct definition for “logically implies” in those approaches that allow “strong generalization” (3.4). In particular, this definition allows that $A \models (\forall x)A$ —as we indeed require in order to “match” the syntactic $A \vdash (\forall x)A$. The flip side of this is that $\models A \Rightarrow B$ and $A \models B$ are *not* equivalent.



4.7 Definition. (*First order theories*) Given a first order language $L = (V, \mathbf{Term}, \mathbf{Wff})$, a (*first order*) *theory* Γ over L consists of all the formulas in Γ taken as *nonlogical* axioms, along with the logical axioms (see 2.5), and the rules of inference **Inf2–Inf4**. **Thm $_{\Gamma}$** are the theorems of the theory.

A *pure theory* is one with $\Gamma = \emptyset$. \square



4.8 Remark. In the next section we will consider two different ways to “do logic”, and therefore will have two kinds of first order theories. One will be as in the definition above. We will call it an “E-theory”, to indicate that the logical axioms and especially the rules of inference support equational reasoning. The other will be as developed in [Sh]. We will call such a theory an “S-theory”. 

4.9 Definition. (*Soundness*) A *pure theory* is *sound*, iff $\vdash A$ implies $\models A$, that is, iff all the theorems of the theory are universally valid. \square

4.10 Metatheorem. *The pure E-theory of definition 4.7 is sound.*

Proof. Let $\vdash A$. Pick an arbitrary structure $\mathfrak{M} = (M, \mathcal{I})$ and do induction on \emptyset -theorems to show that $\models_{\mathfrak{M}} A$.

Basis. A is a logical axiom (see 2.5). Now the closure of axioms in group **Ax1** under **Inf1** is a set of tautologies, hence, by 4.4, if A belongs there, $A^{\mathcal{I}} = \mathbf{t}$.

Note that the *form* of schemata **Ax2–Ax5** is invariant under an application of **Inf1**.

If A is an instance of **Ax3**, then a proof that $\overline{A}^{\mathcal{I}} = \mathbf{t}$ for any \mathfrak{M} -instance \overline{A} of A is easily accessible in the literature (e.g., [En], [Men]; [Sh] handles the dual axiom $B[x := t] \Rightarrow (\exists x)B$, where t is substitutable in x).

The literature also takes care of the equality axioms (**Ax4** and **Ax5**). Let us here look into **Ax2**.

Say A is

$$B \vee (\forall x)C \equiv (\forall x)(B \vee C) \quad (1)$$

where x is *not* free in B . Let

$$\overline{B} \vee (\forall x)\overline{C} \equiv (\forall x)(\overline{B} \vee \overline{C})$$

be an \mathfrak{M} -instance of (1). We need to show that

$$(\overline{B} \vee (\forall x)\overline{C})^{\mathcal{I}} = \mathbf{f} \text{ iff } ((\forall x)(\overline{B} \vee \overline{C}))^{\mathcal{I}} = \mathbf{f} \quad (2)$$

Assume the left side of iff. Then $(\overline{B})^{\mathcal{I}} = \mathbf{f}$ and $((\forall x)\overline{C})^{\mathcal{I}} = \mathbf{f}$. The latter entails $(\overline{C}[x := \hat{i}])^{\mathcal{I}} = \mathbf{f}$, for some $i \in M$. Since x is not free in B , $(\overline{B})^{\mathcal{I}} \vee (\overline{C}[x := \hat{i}])^{\mathcal{I}\dagger}$

[†]The \vee here is the metamathematical truth function on $\{\mathbf{t}, \mathbf{f}\}$.

is the meaning of an \mathfrak{M} -instance of $B \vee C$, and is, of course, \mathbf{f} . Thus, the right side of iff is \mathbf{f} (by 4.4).

The other direction is similar.

We next show that if A is the conclusion of a rule, then it is universally valid if the premises are so. By 3.16, we only look at *Leibniz*.

Leibniz. Let A be $B[p := C] \equiv B[p := D]$ and

$$\models_{\mathfrak{M}} C \equiv D \tag{4}$$

By induction on the formula B , one proves

$$\models_{\mathfrak{M}} B[p := C] \equiv B[p := D] \tag{5}$$

The interesting induction step is when B is $(\forall x)E$, and p occurs in E —but without loss of generality in neither C nor D .

Pick an arbitrary \mathfrak{M} -instance of $E[p := C] \equiv E[p := D]$.

Compare $(\overline{E}[p := \overline{C}])^{\mathcal{I}}$ with $(\overline{E}[p := \overline{D}])^{\mathcal{I}}$. Since, by (4), $\overline{C}^{\mathcal{I}} = \overline{D}^{\mathcal{I}}$, I.H. (on formulas) gives

$$(\overline{E}[p := \overline{C}])^{\mathcal{I}} = (\overline{E}[p := \overline{D}])^{\mathcal{I}}$$

where the “=” above is on $\{\mathbf{t}, \mathbf{f}\}$. By 4.4, we get

$$((\forall x)(\overline{E}[p := \overline{C}]))^{\mathcal{I}} = ((\forall x)(\overline{E}[p := \overline{D}]))^{\mathcal{I}}$$

and we are done with both inductions. \square



A by-product of soundness is consistency. A theory Γ is *consistent* iff $\mathbf{Thm}_{\Gamma} \subset \mathbf{Wff}$ (proper subset).

Thus, the pure E-theory is consistent, since, by soundness, *false* is not provable.

5. Completeness

5.1 Definition. A theory Γ is *complete* iff $\Gamma \models A$ implies $\Gamma \vdash A$ for any formula A . \square

One way to show completeness of the pure E-theory is to offer a proof in the style of Henkin. A much easier way is to show that the pure E-theory is at least as powerful as the pure S-theory.

Shoenfield allows no propositional variables, and admits therefore axiom schemata throughout.

The axioms in Shoenfield are

S-Ax1 The schema $A \vee \neg A$

S-Ax2 The *dual* of **Ax3**, namely, $A[x := t] \Rightarrow (\exists x)A$, where t is substitutable in x

S-Ax3 The E-schemata **Ax4** and **Ax5**.

The rules of inference (after the tautology theorem) boil down to

S-Inf1 Any rule

$$\frac{A_1, \dots, A_n}{B}$$

is permissible as long as $A_1, \dots, A_n \models_{\mathbf{Taut}} B$

S-Inf2

$$\frac{A \Rightarrow B}{(\exists x)A \Rightarrow B}$$

provided x is not free in B .

Let us write $\vdash_{\text{theory}} A$ (where “theory” is “E” or “S”) to indicate what axioms and what rules were responsible for the proof of A . We have at once,

5.2 Metatheorem. *For any A that contains no propositional variables, $\vdash_S A$ implies $\vdash_E A$.*

Proof. The trivial proof on theorems (of S) is omitted. We only note that $\vdash_E A \vee \neg A$ by 3.14. \square

In order to prove next that $\models A$ implies $\vdash_E A$, via S, we need a simple technical Lemma.

5.3 Lemma. *Let A be a formula over the language L of section 1, and let p be a propositional variable that occurs in A .*

Extend the language L by adding P , a new 1-ary predicate symbol.

Then, $\models A$ iff $\models A[p := (\forall x)Px]$ and $\vdash_E A$ iff $\vdash_E A[p := (\forall x)Px]$.

Proof. (\models) The only-if is by soundness (substitution 3.24 was used).

For the if-part pick any structure $\mathfrak{M} = (M, \mathcal{I})$ and let \bar{A} be an \mathfrak{M} -instance of A . Expand \mathfrak{M} to $\mathfrak{M}' = (V, \mathcal{I}')$ where \mathcal{I}' is the same as \mathcal{I} , *except* that it also gives meaning to P as follows: If $p^{\mathcal{I}} = \mathbf{t}$, set $P^{\mathcal{I}'} = M$, else set $P^{\mathcal{I}'} = \emptyset$. Clearly, $p^{\mathcal{I}'} = ((\forall x)Px)^{\mathcal{I}'}$.

By assumption, $\bar{A}[p := (\forall x)Px]^{\mathcal{I}'} = \mathbf{t}$, hence, $\bar{A}^{\mathcal{I}} = \bar{A}^{\mathcal{I}'} = \mathbf{t}$ as well.

(\vdash) The only-if is the result of 3.24.

For the if part, let $\vdash_E A[p := (\forall x)Px]$. By induction on \emptyset -theorems we show that $\vdash_E A$ as well.

Basis. $A[p := (\forall x)Px]$ is a logical axiom (see 2.5). If it is a substitution (**Inf1**) instance of a formula in group **Ax1**, then it is a tautology. But then so is A , hence (by 3.14) $\vdash_E A$. **Ax4** is not applicable (it cannot contain $(\forall x)Px$). It is clear that if $A[p := (\forall x)Px]$ is an instance of one of the schemata **Ax2–Ax3** or **Ax5**, then replacing all occurrences of $(\forall x)Px$ in $A[p := (\forall x)Px]$ by the

propositional variable p results to a formula of the same form, so A is still an axiom, hence $\vdash_E A$.

Leibniz. Let $\vdash_E B \equiv C$ and $A[p := (\forall x)Px]$ be $D[q := B] \equiv D[q := C]$. Let B', C', D' be obtained from B, C, D by replacing every occurrence of $(\forall x)Px$ by p .

By I.H., $\vdash_E B' \equiv C'$, hence (Leibniz) $\vdash_E D'[q := B'] \equiv D'[q := C']$, i.e., $\vdash_E A$.

$\models_{\mathbf{Taut}}$. Let

$$B_1, \dots, B_r \models_{\mathbf{Taut}} A[p := (\forall x)Px] \quad (1)$$

and $\vdash_E B_i$ ($i = 1, \dots, r$). Clearly (1) is only dependent on the configuration of the propositional connectives, hence $B'_1, \dots, B'_r \models_{\mathbf{Taut}} A$, where B'_i is obtained from B_i by replacing all occurrences of $(\forall x)Px$ by p .

Hence $\vdash_E A$. \square

5.4 Corollary. *The pure E-theory of definition 4.7 is complete.*

Proof. Let $\models A$. Let A' be obtained from A by replacing each occurrence of *true* (respectively *false*) by $p \vee \neg p$ (respectively $p \wedge \neg p$) where p is a propositional variable not occurring in A . Let A'' be obtained from A' by replacing each propositional variable p, q, \dots in it by $(\forall x)Px, (\forall x)Qx, \dots$ respectively, where P, Q, \dots are *new* predicate symbols.

Clearly, by 5.3, $\models A''$. This formula is in the language of S. Thus, by completeness of S, $\vdash_S A''$.

By 5.2, $\vdash_E A''$, hence $\vdash_E A'$, by 5.3.

Finally, by Leibniz, $\vdash_E A$. \square



Completeness and soundness were proved for a pure “E-theory” that has a different (much smaller) axiom set than the one given in [GS1]. The next section shows the equivalence of the present pure theory with the one offered in [GS1].



6. Comparison with the Equational Logic of [GS1]

We prove that each rule or axiom on quantification that is offered in [GS1], chapters 8 and 9, is “derived” (hence redundant or dependent) in the E-logic that we have presented here (definitions 2.1, 2.3 and 2.5). We write \vdash meaning \vdash_E throughout.

Nomenclature and numbers given in brackets are those in [GS1].

A.1 “Leibniz (8.12)” The following two rules are meant to supplement the “propositional” version given in [GS1], towards predicate calculus duty ([GS1], p.148).

$$\frac{A \equiv B}{(\forall x)(C[p := A] \Rightarrow D) \equiv (\forall x)(C[p := B] \Rightarrow D)}$$

and

$$\frac{D \Rightarrow (A \equiv B)}{(\forall x)(D \Rightarrow C[p := A]) \equiv (\forall x)(D \Rightarrow C[p := B])} \quad (1)$$

The first is a direct application of **Inf2**, where, without loss of generality, p occurs only in C . (If not, apply **Inf2** to $C' \Rightarrow D$ instead, where C' is $C[p := q]$ — q a propositional variable not in $C \Rightarrow D$. Be sure to use $q := A$ and $q := B$ in the application of **Inf2**).

(1) is *not* valid, unless we drop “ $D \Rightarrow$ ” on the premise side (should read just $A \equiv B$), in which case it follows from **Inf2** exactly like the first of the two (8.12)-rules.

Indeed, take D to be $x = 0$, C to be $(\forall x)p$, A to be $x = 0$ and B to be *true*. Then,

$$\models D \Rightarrow (A \equiv B)$$

that is

$$\models x = 0 \Rightarrow (x = 0 \equiv \text{true})$$

but

$$\not\models (\forall x)(D \Rightarrow C[p := A]) \equiv (\forall x)(D \Rightarrow C[p := B])$$

that is

$$\not\models (\forall x)(x = 0 \Rightarrow (\forall x)x = 0) \equiv (\forall x)(x = 0 \Rightarrow (\forall x)\text{true})$$

A.2 “Empty range (8.13)”. $\vdash (\forall x)(\text{false} \Rightarrow A) \equiv \text{true}$. It follows from 3.2, $\vdash \text{false} \Rightarrow A \equiv \text{true}$, and **Inf2**.

A.3 “One-point rule (8.14)”. Provided that x is not free in t , and t is substitutable in x in A ,[†] $\vdash (\forall x)(x = t \Rightarrow A) \equiv A[x := t]$.

(\Rightarrow) By **Ax3**, $\vdash (\forall x)(x = t \Rightarrow A) \Rightarrow t = t \Rightarrow A[t]$. By 3.5 and **Ax4**, $\vdash t = t$, hence $\vdash (\forall x)(x = t \Rightarrow A) \Rightarrow A[t]$ by 3.15.

(\Leftarrow) A standard result (from **Ax4**, **Ax5**) is that $\vdash x = y \Rightarrow y = x$ (e.g., [Sh]). By 3.5, $\vdash x = t \Rightarrow t = x$, thus (3.15 and **Ax5**) $\vdash A[t] \Rightarrow x = t \Rightarrow A$. As x is not free in $A[t]$, 3.8 yields $\vdash A[t] \Rightarrow (\forall x)(x = t \Rightarrow A)$.

A.4 “Distributivity (8.15)”.

$$\vdash (\forall x)(A \Rightarrow B) \wedge (\forall x)(A \Rightarrow C) \equiv (\forall x)(A \Rightarrow B \wedge C).$$

[†]The substitutability condition is necessary, but not mentioned in [GS1]—see 2.2(3).

(\Rightarrow) Add $(\forall x)(A \Rightarrow B) \wedge (\forall x)(A \Rightarrow C)$ as a (nonlogical) axiom. Then,

1. $(\forall x)(A \Rightarrow B) \wedge (\forall x)(A \Rightarrow C)$
2. $(\forall x)(A \Rightarrow B)$ ⟨1. and 3.15⟩
3. $(\forall x)(A \Rightarrow C)$ ⟨1. and 3.15⟩
4. $A \Rightarrow B$ ⟨2. and *specialization*⟩
5. $A \Rightarrow C$ ⟨3. and *specialization*⟩
6. $A \Rightarrow B \wedge C$ ⟨4., 5. and 3.15⟩
7. $(\forall x)(A \Rightarrow B \wedge C)$ ⟨6. and 3.4⟩

By the Deduction Theorem, we are done.

(\Leftarrow) With amended “hints”, the above proof can be reversed (7.–1.)

A.5 (8.16)–(8.18) in [GS1] boil down to (8.18) if “*” is “ \forall ”.

“Range split (8.18)”. $(\forall x)(A \vee B \Rightarrow C) \equiv (\forall x)(A \Rightarrow C) \wedge (\forall x)(B \Rightarrow C)$.

(\Rightarrow) Add $(\forall x)(A \vee B \Rightarrow C)$. Then,

1. $(\forall x)(A \vee B \Rightarrow C)$
2. $A \vee B \Rightarrow C$ ⟨1. and *specialization*⟩
3. $A \Rightarrow C$ ⟨2. and 3.15⟩
4. $B \Rightarrow C$ ⟨2. and 3.15⟩
5. $(\forall x)(A \Rightarrow C)$ ⟨3. and 3.4⟩
6. $(\forall x)(B \Rightarrow C)$ ⟨4. and 3.4⟩
7. $(\forall x)(A \Rightarrow C) \wedge (\forall x)(B \Rightarrow C)$ ⟨5., 6. and 3.15⟩

By the Deduction Theorem, we are done.

(\Leftarrow) Reverse the above proof.

A.6 “Interchange of dummies (8.19)”.

$(\forall x)(A \Rightarrow (\forall y)(B \Rightarrow C)) \equiv (\forall y)(B \Rightarrow (\forall x)(A \Rightarrow C))$, on the condition that y is not free in A and x is not free in B .

(\Rightarrow) Add $(\forall x)(A \Rightarrow (\forall y)(B \Rightarrow C))$. Then,

1. $(\forall x)(A \Rightarrow (\forall y)(B \Rightarrow C))$
2. $A \Rightarrow (\forall y)(B \Rightarrow C)$ ⟨1. and *specialization*⟩
3. $A \Rightarrow B \Rightarrow C$ ⟨2. and 3.8; y not free in A ⟩
4. $B \Rightarrow A \Rightarrow C$ ⟨3. and 3.15⟩
5. $B \Rightarrow (\forall x)(A \Rightarrow C)$ ⟨4. and 3.8; x not free in B ⟩
6. $(\forall y)(B \Rightarrow (\forall x)(A \Rightarrow C))$ ⟨5. and 3.4⟩

By the Deduction Theorem, we are done.

(\Leftarrow) Reverse the above proof.

A.7 “Nesting (8.20)”. $(\forall x)(\forall y)(A \wedge B \Rightarrow C) \equiv (\forall x)(A \Rightarrow (\forall y)(B \Rightarrow C))$, on the condition that y is not free in A .

(\Rightarrow) Add $(\forall x)(\forall y)(A \wedge B \Rightarrow C)$. Then,

1. $(\forall x)(\forall y)(A \wedge B \Rightarrow C)$
2. $A \wedge B \Rightarrow C$ (1. and *specialization*)
3. $A \Rightarrow B \Rightarrow C$ (2. and 3.15)
4. $A \Rightarrow (\forall y)(B \Rightarrow C)$ (3. and 3.8; y not free in A)
5. $(\forall x)(A \Rightarrow (\forall y)(B \Rightarrow C))$ (4. and 3.4)

By the Deduction Theorem, we are done.

(\Leftarrow) Reverse the above steps.

A.8 “Dummy renaming (8.21)”. This is 3.10.

A.9 “Trading (9.2)”. This is an abbreviation in the metalanguage. $(\forall x)_B$ is not a “new quantifier”. Thus, $((\forall x)_B A)$ abbreviates $((\forall x)(B \Rightarrow A))$.

A.10 “Distributivity of \vee over \forall (9.5)”. Included as **Ax2**.

This concludes that for every formula A , $\vdash_{GS1} A$ implies $\vdash_E A$, where \vdash_{GS1} indicates provability in the system outlined in [GS1].

Conversely, the set of axioms in [GS1]—along with the (corrected) “augmented Leibniz” (8.12)—subsume our version of pure E-theory (our **Ax3** is a theorem in [GS1]). Thus,

6.1 Metatheorem. For every formula A , $\vdash_{GS1} A$ iff $\vdash_E A$.

6.2 Corollary. The pure first order theory outlined in [GS1] is sound and complete.

7. Concluding remarks; an alternative formalism for Equational Logic

We have offered a careful formalization of the Equational Predicate Logic outlined in [GS1] and we have proved that this formalization is sound and complete.

We want to briefly investigate here an alternative formalization of Equational Logic, based on a weaker form of the Leibniz rule. We have already observed

(see the remark following 3.4) that since **Inf2** yields “strong generalization”, first order Logics such as [Bou, En] do *not* support it.

We show how such Logics can be recast in the Equational Logic paradigm.

We base our discussion here, for the sake of concreteness, to the version of first order Logic as it is defined in [En]. We start by describing this logic.

We allow propositional variables in formulas only for the purpose of making the statement of “Leibniz”, below, easy. The only connectives are $\neg, \Rightarrow, \forall$. The remaining connectives— $\equiv, \vee, \wedge, \exists$ —are introduced *as abbreviations*, and the same holds for the constants (0-ary propositional connectives in reality) *true* and *false* (for example, in view of the theorem $\vdash (A \equiv A) \equiv (B \equiv B)$, we abbreviate any occurrence of $A \equiv A$ by *true*).

There is only one rule of inference, *modus ponens*.

The axioms (*schemata*) are all the possible “partial” generalizations[†] of the following

En1. All tautologies (in the sense of 3.12),

En2. All formulas of the form $(\forall x)A \Rightarrow A[t]$, *provided* t is substitutable in x ,

En3. All formulas of the form $(\forall x)(A \Rightarrow B) \Rightarrow (\forall x)A \Rightarrow (\forall x)B$ (see 3.18),

En4. All formulas of the form $A \Rightarrow (\forall x)A$, *provided* x is not free in A (see 3.7).

En5. Ax4 and Ax5.

In this logic the following weaker version of **Inf2**, let us call it *no-capture Leibniz*, is a derived rule:

$$\frac{A \equiv B}{C[p := A] \equiv C[p := B]}$$

provided neither A nor B contain free variables that are captured by quantifiers during the above substitutions.

For a proof, we note that the following Metatheorem (“restricted generalization”) holds in [En]:

$\Gamma \vdash G$ iff $\Gamma \vdash (\forall x)G$, *provided no formula in Γ has x free*. Here then is the proof of *no-capture Leibniz*, $A \equiv B \vdash C[p := A] \equiv C[p := B]$, by induction on the formula C .

Basis. C is atomic. If C is the string p , then $C[p := A]$ is A and $C[p := B]$ is B , so our conclusion is our hypothesis.

In all other cases $C[p := A] \equiv C[p := B]$ is the tautology $C \equiv C$.

Induction Step(s) (I.S.).

I.S.1. C is $\neg D$. By I.H., $A \equiv B \vdash D[p := A] \equiv D[p := B]$.

Since $D[p := A] \equiv D[p := B] \vdash_{\mathbf{Taut}} \neg D[p := A] \equiv \neg D[p := B]$, we are done in this case.[†]

[†]A generalization B of A is *partial* iff B is A , prefixed with zero or more strings $(\forall x)$ — x may or may not occur in A . The well known “universal closure of A ” is a special case.

[†]The tautology theorem holds in [En].

I.S.2. C is $D \Rightarrow G$. By I.H., $A \equiv B \vdash D[p := A] \equiv D[p := B]$ and $A \equiv B \vdash G[p := A] \equiv G[p := B]$. Since $D[p := A] \equiv D[p := B]$, $G[p := A] \equiv G[p := B] \vdash_{\mathbf{Taut}} (D \Rightarrow G)[p := A] \equiv (D \Rightarrow G)[p := B]$, we are done once more.

I.S.3. C is $(\forall x)D$. By I.H., $A \equiv B \vdash D[p := A] \equiv D[p := B]$, hence (Tautology Theorem) $A \equiv B \vdash D[p := A] \Rightarrow D[p := B]$.

By *restricted generalization*, since x is not free in $A \equiv B$, $A \equiv B \vdash (\forall x)(D[p := A] \Rightarrow D[p := B])$, hence by modus ponens and **En3**, $A \equiv B \vdash (\forall x)D[p := A] \Rightarrow (\forall x)D[p := B]$.

Similarly (from $A \equiv B \vdash D[p := A] \Leftarrow D[p := B]$), $A \equiv B \vdash (\forall x)D[p := A] \Leftarrow (\forall x)D[p := B]$, hence, one more application of the Tautology Theorem gives $A \equiv B \vdash (\forall x)D[p := A] \equiv (\forall x)D[p := B]$. This concludes the proof.

Assume now that we want to do Equational Logic, based on no-capture Leibniz.

We adopt **En1–En5** above, but our rules—instead of modus ponens—are **Inf3**, **Inf4** and no-capture Leibniz.

Trivially, modus ponens is a derived rule, and thus [En]-logic and this version of Equational Logic are equivalent. *In particular, the latter is sound and complete.*

This study also demonstrates, via an entirely different avenue than the one taken in [GS3], that “full Leibniz” (**Inf2**) is strictly more powerful than the “no-capture” version *even in a complete logic*. The similar result in [GS3] is shown for an incomplete logic, one in which **Ax3** is not deducible.

There are many ways to “massage” the rule Leibniz. We wish to cite here two more versions and gauge their relative power.

First, let us restrict the rule on *universally valid* formulas only, *whenever “capture” occurs in its application*. Syntactically, this is rule $L2$ below.

$\Gamma \vdash A \equiv B$ implies $\Gamma \vdash C[p := A] \equiv C[p := B]$ *provided* $\Gamma = \emptyset$ if capture occurs
($L2$)

The same proof we presented above for the no-capture version (in the logic of [En]) can be re-traced with trivial modifications to yield $L2$ as a derived rule (restricted generalization with $\Gamma = \emptyset$ is used).

Conversely, adopting **En1–En5** above, and the rules—instead of modus ponens—**Inf3**, **Inf4** and $L2$, we can derive modus ponens:

- | | |
|-------------------------|---------------------------------------|
| 1. A | ⟨ given ⟩ |
| 2. $A \Rightarrow B$ | ⟨ given ⟩ |
| 3. $true \Rightarrow B$ | ⟨ 1., 2., 3.1, $L2$ and Inf3 ⟩ |
| 4. B | ⟨ 3. and 3.15 ⟩ |

Thus, in the presence of **En1–En5**, and **Inf3**, **Inf4**, $L2$ and no-capture Leibniz are equivalent, and the system so founded is equivalent to that in [En].

Indeed, the proof of modus ponens above makes it clear that all we really need is a *propositional Leibniz*, PL , that is

$$\frac{A \equiv B}{C[p := A] \equiv C[p := B]}, \text{ where } p \text{ is not in the scope of a quantifier} \quad (PL)$$

In an application of PL , we *do not* require that $\vdash A \equiv B$.

We can summarize this discussion thus:

7.1 Metatheorem. *In the presence of the axioms **En1–En5** and the rules **Inf3**, **Inf4**, all the following versions of the Leibniz rule are equivalent: PL , “no-capture”, and $L2$. Adopting any of them yields a sound and complete first order logic (equivalent to that in $[En]$).*

*In this logic, “full” Leibniz (**Inf2**) is not derivable.*

An instructor teaching Equational Logic to first year students will probably be tempted to “interpret” $[GS1]$ logic in this simpler framework—using PL and demonstrating that both $L2$ and no-capture Leibniz are derived rules.

This alternative version may well prove more palatable in the classroom, if for no other reason, because of the simplicity of its rules of inference and its support for an unconstrained Deduction Theorem.

Acknowledgements. I warmly thank my colleague Steve Watson who carefully read a previous version of the paper and offered valuable advice. In particular he pointed to a problematic use of **Inf1** in my original definition 2.5, which I have now fixed. I am also indebted to my colleague Jonathan Ostroff who encouraged me to simplify the formalism. Heeding his request I came up with the simpler formalism ($[En]$ -based) of section 7.

8. Bibliography

- [Ba] Barwise, J. “An introduction to first-order logic”, in *Handbook of Mathematical Logic* (J. Barwise, Ed.), 5–46, Amsterdam: North-Holland Publishing Company, 1978.
- [Bou] Bourbaki, N. *Éléments de Mathématique; Théorie des Ensembles*, **Ch. 1**, Paris: Hermann, 1966.
- [DSc] Dijkstra, E. W., and Scholten, C. S. B. *Predicate Calculus and Program Semantics*, New York: Springer-Verlag, 1990.
- [En] Enderton, H. B. *A mathematical introduction to logic*, New York: Academic Press, 1972.
- [GS1] Gries, D. and Schneider, F. B. *A Logical Approach to Discrete Math*, New York: Springer-Verlag, 1994.

- [GS2] Gries, D. and Schneider, F. B. *Equational propositional logic*, Information Processing Letters, **53**, 145–152, 1995.
- [GS3] Gries, D. and Schneider, F. B. *Formalizations of substitution of equals for equals*, Pre-print, Sept. 1998 (personal communication).
- [Man] Manin, Yu. I. *A Course in Mathematical Logic*, New York: Springer-Verlag, 1977.
- [Men] Mendelson, E. *Introduction to mathematical logic*, 3rd Edition, Monterey, Calif: Wadsworth & Brooks, 1987.
- [Sh] Shoenfield, J. R. *Mathematical Logic*, Reading, Mass.: Addison-Wesley, 1967.